ARCOS – Post-Processor Guide

1. Purpose of this Guide

This guide explains the role of the Post-Processor in ARCOS (AI Rule-Constrained Orchestration System). It is written for developers, clarifying that the Post-Processor is a process with defined boundaries. Developers can implement it however they wish, provided that schema-based communication protocols are respected. The Post-Processor must handle specific inputs and produce specific outputs, but internal design is entirely flexible.

2. Role in ARCOS

The Post-Processor refines (or compiles in the BLEU example) validated outputs from the Producer, preparing final artifacts for use.

3. Responsibilities

- Take validated outputs from the Producer and finalize them (e.g., compile, optimize, package, format).
- Return PostProcessorResponse.xml with success or failure status.
- Produce PostProcessorReport.xml containing compiler diagnostics, build artifacts, or summaries.

4. Workflow

1. Receive PostProcessorRequest from the Orchestrator with ProducerOutput and checksum.
2. Run post-processing tasks (e.g., compilation, packaging, transformation).
3. Return PostProcessorResponse.xml (Success or Failure).
4. Produce PostProcessorReport.xml with diagnostics and artifacts if applicable.

5. Inputs and Outputs

- Inputs: ProducerOutput (with checksum).
- Outputs: PostProcessorResponse.xml (success/failure), PostProcessorReport.xml (diagnostics and artifact list).

6. Example (BLEU Inventory Domain)

In the BLEU domain, the Post-Processor compiles the Rust CRUD code generated by the Producer, returning a report with build logs, artifact paths, and success/failure status.

7. Benefits for Developers

- Implementation freedom: any compiler, packager, or pipeline can be used.
- Supports detailed reporting tailored to the domain.
- Encourages modular separation of code generation (Producer) from code finalization (Post-Processor).

8. Conclusion

The Post-Processor is a well-defined ARCOS component: it operates within schema-constrained boundaries, while giving developers freedom of implementation. By respecting the communication contracts, developers ensure that their {title} agent integrates seamlessly into ARCOS flows.